

DBMS Assignment No.8

```
import java.awt.*;
import java.awt.event.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.swing.*;

public class student extends JFrame implements
ActionListener
{
    JFrame f;
    JLabel l1, l2,l3,l4;
    JTextField t1, t2,t3;
    JButton b1, b2, b3, b4, b5;
    Connection c;
    Statement s;
    ResultSet r;
    student ()
    {
        try
        {
            f=new JFrame("Student Form");
            f.setLayout(null);f.setVisible(true);
            f.setSize(700, 500);
            l4=new JLabel("Student Management System");
            //l4.setBounds(100,01,250,250);
            l4.setBounds(100, 30, 400, 30);
            f.add(l4);
            l4.setForeground(Color.blue);
            l4.setFont(new Font("Serif", Font.BOLD,30));
            l1=new JLabel("Stud_RollNo");
            l1.setBounds(50, 70, 100, 50);
            f.add(l1);
            l2=new JLabel("Stud_Name");
            l2.setBounds(50, 120, 100, 50);
            f.add(l2);
            l3=new JLabel("Stud_Dept");
            l3.setBounds(50, 170, 100, 50);
            f.add(l3);
            t1=new JTextField();
            t1.setBounds(150, 90, 100, 30);
            f.add(t1);
            t2=new JTextField();
            t2.setBounds(150, 140, 100, 30);
            f.add(t2);t3=new JTextField();
            t3.setBounds(150, 190, 100, 30);
            f.add(t3);
            b1= new JButton("ADD");
            b1.setBounds(200, 300, 75, 50);
            f.add(b1);
            b1.addActionListener(this);
            b2= new JButton("EDIT");
```

```

        b2.setBounds(300, 300, 75, 50);
        f.add(b2);
        b2.addActionListener(this);
        b3= new JButton("DELETE");
        b3.setBounds(400, 300, 75, 50);
        f.add(b3);
        b3.addActionListener(this);
        b5= new JButton("EXIT");
        b5.setBounds(500, 300, 75, 50);
        f.add(b5);
        b5.addActionListener(this);
        Class.forName("com.mysql.jdbc.Driver");
        c=DriverManager.getConnection("jdbc:mysql://localhost:3306/info","root","root");s=c.createStatement();
    }

    catch(Exception e)
    {
        System.out.println(e);
    }
} //ends INS Constructor

public void actionPerformed(ActionEvent ae)
{
    try
    {
        if(ae.getSource()==b1)
        {
            String s1="INSERT
            INTO result(stud_RollNo,stud_Name,stud_Dept)
            VALUES('"+t1.getText()+"','"+t2.getText()+"','"+t3.getText()+"');
            System.out.println(s1);
            s.executeUpdate(s1);
            r=s.executeQuery("SELECT * FROM result");
            t1.setText("");
            t2.setText("");
            t3.setText("");
        }
        else if(ae.getSource()==b2)
        {
            String s2="UPDATE user1 SET
            stud_Name='"+t2.getText()+"' WHERE
            stud_RollNo='"+t1.getText()+"';
            System.out.println(s2);
            s.executeUpdate(s2);
            r=s.executeQuery("SELECT * FROM result");
            t1.setText("");
            t2.setText("");t3.setText("");
        }
        else if(ae.getSource()==b3)
        {
            String s3="DELETE FROM result WHERE
            stud_RollNo='"+t1.getText()+"';
            System.out.println(s3);
            s.executeUpdate(s3);
        }
    }
}

```

```

        r=s.executeQuery("SELECT * FROM result");
        t1.setText("");
        t2.setText("");
        t3.setText("");
    }
    else if(ae.getSource()==b5)
    {
        System.exit(0);
    }
}

catch(Exception e)
{
    System.out.println(e);
}
}

public static void main(String args[])
{
    new student();
}
}

```

Output:

```

sl2-pc5@sl2pc5-HP-Compaq-4000-Pro-SFF-PC:~$
mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with
; or \g.
Your MySQL connection id is 42
Server version: 5.5.61-0ubuntu0.14.04.1 (Ubuntu)
Copyright (c) 2000, 2018, Oracle and/or its
affiliates. All rights reserved.
Oracle is a registered trademark of Oracle
Corporation and/or its affiliates. Other names may be trademarks of
their respective
owners.
Type 'help;' or '\h' for help. Type '\c' to
clear the current input statement.

```

```

mysql> create database info;
Query OK, 1 row affected (0.03 sec)

```

```

mysql> use info;
Database changed

```

```

mysql> create table result (stud_RollNo int,stud_Name varchar(20),stud_Dept
varchar(20));
Query OK, 0 rows affected (0.08 sec)

```

```
mysql> select *from result;
+-----+-----+-----+
| stud_RollNo| stud_Name| stud_Dept|
+-----+-----+-----+
|          1 |      abc |      comp |
+-----+-----+-----+
1 row in set (0.00 sec)
```

//ADD DATA

```
mysql> select *from result;
+-----+-----+-----+
| stud_RollNo| stud_Name| stud_Dept|
+-----+-----+-----+
|          1 |      abc |      comp |
|          2 |    harsha |      comp |
|          3 |      tej |      comp |
|          4 |      rina |      mech |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

//DELETE DATA

```
mysql> select *from result;
+-----+-----+-----+
| stud_RollNo| stud_Name| stud_Dept|
+-----+-----+-----+
|          2 |    harsha |      comp |
|          3 |      tej |      comp |
|          4 |      rina |      mech |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

DBMS Assignment No.9

Please enter a MongoDB connection string (Default: mongodb://localhost/): *mongodb:mongodb:*
Current Mongosh Log ID: 651a4730afa7b5d7be217232
Connecting to:
mongodb://127.0.0.1:27017/mongodb?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.10.6
Using MongoDB: 7.0.1 Using Mongosh: 1.10.6
mongosh 2.0.1 is available for download: <https://www.mongodb.com/try/download/shell>
For mongosh info see: <https://docs.mongodb.com/mongosh-shell/>

The server generated these startup warnings when booting
2023-09-30T22:09:56.935+05:30: Access control is **not** enabled for the database.
Read and write access to data and configuration is unrestricted

mongodb> use AK
switched to db AK

```
AK> db.createCollection('Student');
{ ok: 1 }
AK> db.Student.insertMany([ { 'Rno': '1', 'Name': 'Atharva', 'Class': 'TE Computer' },
    { 'Rno': '2', 'Name': 'Shubham', 'Class': 'TE Computer' },
    { 'Rno': '3', 'Name': 'Yash', 'Class': 'TE Computer' },
    { 'Rno': '4', 'Name': 'Nikita', 'Class': 'TE Computer' },
    { 'Rno': '5', 'Name': 'Nikhil', 'Class': 'TE Computer' },
    { 'Rno': '6', 'Name': 'Nikhil', 'Class': 'TE Computer' } ]]);
```

```
{
  acknowledged: true,
  insertedIds:
    { '0': ObjectId("651a47c4afa7b5d7be217233"),
      '1': ObjectId("651a47c4afa7b5d7be217234"),
      '2': ObjectId("651a47c4afa7b5d7be217235"),
      '3': ObjectId("651a47c4afa7b5d7be217236"),
      '4': ObjectId("651a47c4afa7b5d7be217237"),
      '5': ObjectId("651a47c4afa7b5d7be217238")
    }
}
```

```
AK> db.Student.find();
[
  {
    _id: ObjectId("651a47c4afa7b5d7be217233"),
    Rno: '1',
    Name: 'Atharva',
    Class: 'TE Computer'
  },
  {
    _id: ObjectId("651a47c4afa7b5d7be217234"),
    Rno: '2',
    Name: 'Shubham',
    Class: 'TE Computer'
  },
  {
    _id: ObjectId("651a47c4afa7b5d7be217235"),
    Rno: '3',
    Name: 'Yash',
    Class: 'TE Computer'
  },
  {
    _id: ObjectId("651a47c4afa7b5d7be217236"),
    Rno: '4',
    Name: 'Nikita',
    Class: 'TE Computer'
  },
  {
    _id: ObjectId("651a47c4afa7b5d7be217237"),
    Rno: '5',
    Name: 'Nikhil',
    Class: 'TE Computer'
  },
  {
    _id: ObjectId("651a47c4afa7b5d7be217238"),
    Rno: '6',
    Name: 'Nikhil',
    Class: 'TE Computer'
  }
]
```

```

{
  _id: ObjectId("651a47c4afa7b5d7be217235"),
  Rno: '3',
  Name: 'Yash',
  Class: 'TE Computer'
},
{
  _id: ObjectId("651a47c4afa7b5d7be217236"),
  Rno: '4',
  Name: 'Nikita',
  Class: 'TE Computer'
},
{
  _id: ObjectId("651a47c4afa7b5d7be217237"),
  Rno: '5',
  Name: 'Nikhil',
  Class: 'TE Computer'
},
{
  _id: ObjectId("651a47c4afa7b5d7be217238"),
  Rno: '6',
  Name: 'Nikhil',
  Class: 'TE Computer'
}
]

```

AK> db.Student.find().pretty();

```

[
  {
    _id: ObjectId("651a47c4afa7b5d7be217233"),
    Rno: '1',
    Name: 'Atharva',
    Class: 'TE Computer'
  },
  {
    _id: ObjectId("651a47c4afa7b5d7be217234"),
    Rno: '2',
    Name: 'Shubham',
    Class: 'TE Computer'
  },
  {
    _id: ObjectId("651a47c4afa7b5d7be217235"),
    Rno: '3',
    Name: 'Yash',
    Class: 'TE Computer'
  },
  {
    _id: ObjectId("651a47c4afa7b5d7be217236"),
    Rno: '4',
    Name: 'Nikita',
    Class: 'TE Computer'
  },
  {
    _id: ObjectId("651a47c4afa7b5d7be217237"),
    Rno: '5',

```

```

    Name: 'Nikhil',
    Class: 'TE Computer'
  },
  {
    _id: ObjectId("651a47c4afa7b5d7be217238"),
    Rno: '6',
    Name: 'Nikhil',
    Class: 'TE Computer'
  }
]

```

```

AK> db.Student.update({'Name':'Atharva'},{$set: {'Name':'Shivraj'}});
DeprecationWarning: Collection.update() is deprecated.
Use updateOne, updateMany, or bulkWrite.

```

```

{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}

```

```

AK> db.Student.find().pretty();

```

```

[
  {
    _id: ObjectId("651a47c4afa7b5d7be217233"),
    Rno: '1',
    Name: 'Shivraj',
    Class: 'TE Computer'
  },
  {
    _id: ObjectId("651a47c4afa7b5d7be217234"),
    Rno: '2',
    Name: 'Shubham',
    Class: 'TE Computer'
  },
  {
    _id: ObjectId("651a47c4afa7b5d7be217235"),
    Rno: '3',
    Name: 'Yash',
    Class: 'TE Computer'
  },
  {
    _id: ObjectId("651a47c4afa7b5d7be217236"),
    Rno: '4',
    Name: 'Nikita',
    Class: 'TE Computer'
  },
  {
    _id: ObjectId("651a47c4afa7b5d7be217237"),
    Rno: '5',
    Name: 'Nikhil',
    Class: 'TE Computer'
  },
]

```

```

{
  _id: ObjectId("651a47c4afa7b5d7be217238"),
  Rno: '6',
  Name: 'Nikhil',
  Class: 'TE Computer'
}
]

```

```

AK> db.Student.deleteOne({'Rno':'6'});
{ acknowledged: true, deletedCount: 1 }

```

```

AK> db.Student.find().pretty();
[
  {
    _id: ObjectId("651a4d4fe64b91d30b77bd80"),
    Rno: '1',
    Name: 'Atharva',
    Class: 'TE Computer'
  },
  {
    _id: ObjectId("651a4d4fe64b91d30b77bd81"),
    Rno: '2',
    Name: 'Shubham',
    Class: 'TE Computer'
  },
  {
    _id: ObjectId("651a4d4fe64b91d30b77bd82"),
    Rno: '3',
    Name: 'Yash',
    Class: 'TE Computer'
  },
  {
    _id: ObjectId("651a4d4fe64b91d30b77bd83"),
    Rno: '4',
    Name: 'Nikita',
    Class: 'TE Computer'
  },
  {
    _id: ObjectId("651a4e7fe64b91d30b77bd86"),
    Rno: '5',
    Name: 'Nikhil',
    Class: 'TE Computer'
  }
]

```

```

AK> db.Student.insertOne({'Rno':'6','Name':'Nikhil','Class':'TE COMPUTER'});
{
  acknowledged: true,
  insertedId: ObjectId("65212b5cab4dfc111e000350")
}

```

```

AK> db.Student.find().pretty();
[
  {
    _id: ObjectId("651a4d4fe64b91d30b77bd80"),

```



```

    Rno: '1',
    Name: 'Atharva',
    Class: 'TE Computer'
  },
  {
    _id: ObjectId("651a4d4fe64b91d30b77bd81"),
    Rno: '2',
    Name: 'Shubham',
    Class: 'TE Computer'
  },
  {
    _id: ObjectId("651a4d4fe64b91d30b77bd82"),
    Rno: '3',
    Name: 'Yash',
    Class: 'TE Computer'
  },
  {
    _id: ObjectId("651a4d4fe64b91d30b77bd83"),
    Rno: '4',
    Name: 'Nikita',
    Class: 'TE Computer'
  },
  {
    _id: ObjectId("651a4e7fe64b91d30b77bd86"),
    Rno: '5',
    Name: 'Nikhil',
    Class: 'TE Computer'
  },
  {
    _id: ObjectId("65212b5cab4dfc111e000350"),
    Rno: '6',
    Name: 'Nikhil',
    Class: 'TE COMPUTER'
  }
]

AK> db.Student.find({$and:[{'Name':'Nikita'},{'Rno':'4'}]});
[
  {
    _id: ObjectId("651a4d4fe64b91d30b77bd83"),
    Rno: '4',
    Name: 'Nikita',
    Class: 'TE Computer'
  }
]

AK> db.Student.find({$or:[{'Name':'Nikita'},{'Rno':'3'}]}).pretty();
[
  {
    _id: ObjectId("651a4d4fe64b91d30b77bd82"),
    Rno: '3',
    Name: 'Yash',
    Class: 'TE Computer'
  },
  {

```

```

    _id: ObjectId("651a4d4fe64b91d30b77bd83"),
    Rno: '4',
    Name: 'Nikita',
    Class: 'TE Computer'
  }
]

AK> db.Student.find({$nor:[{'Name':'Nikita'},{'Rno':'3'}]}).pretty();
[
  {
    _id: ObjectId("651a4d4fe64b91d30b77bd80"),
    Rno: '1',
    Name: 'Atharva',
    Class: 'TE Computer'
  },
  {
    _id: ObjectId("651a4d4fe64b91d30b77bd81"),
    Rno: '2',
    Name: 'Shubham',
    Class: 'TE Computer'
  },
  {
    _id: ObjectId("651a4e7fe64b91d30b77bd86"),
    Rno: '5',
    Name: 'Nikhil',
    Class: 'TE Computer'
  },
  {
    _id: ObjectId("65212b5cab4dfc111e000350"),
    Rno: '6',
    Name: 'Nikhil',
    Class: 'TE COMPUTER'
  }
]

AK> db.Student.find( {"Rno": { $not:{$lt:"3"}}}).pretty();
[
  {
    _id: ObjectId("651a4d4fe64b91d30b77bd82"),
    Rno: '3',
    Name: 'Yash',
    Class: 'TE Computer'
  },
  {
    _id: ObjectId("651a4d4fe64b91d30b77bd83"),
    Rno: '4',
    Name: 'Nikita',
    Class: 'TE Computer'
  },
  {
    _id: ObjectId("651a4e7fe64b91d30b77bd86"),
    Rno: '5',
    Name: 'Nikhil',
    Class: 'TE Computer'
  },
]

```

```

{
  _id: ObjectId("65212b5cab4dfc111e000350"),
  Rno: '6',
  Name: 'Nikhil',
  Class: 'TE COMPUTER'
}
]

```

AK> db.Student.find({"Rno": { \$eq:"5"}}).pretty();

```

[
  {
    _id: ObjectId("651a4e7fe64b91d30b77bd86"),
    Rno: '5',
    Name: 'Nikhil',
    Class: 'TE Computer'
  }
]

```

AK> db.Student.find({"Rno": { \$ne:"5"}}).pretty();

```

[
  {
    _id: ObjectId("651a4d4fe64b91d30b77bd80"),
    Rno: '1',
    Name: 'Atharva',
    Class: 'TE Computer'
  },
  {
    _id: ObjectId("651a4d4fe64b91d30b77bd81"),
    Rno: '2',
    Name: 'Shubham',
    Class: 'TE Computer'
  },
  {
    _id: ObjectId("651a4d4fe64b91d30b77bd82"),
    Rno: '3',
    Name: 'Yash',
    Class: 'TE Computer'
  },
  {
    _id: ObjectId("651a4d4fe64b91d30b77bd83"),
    Rno: '4',
    Name: 'Nikita',
    Class: 'TE Computer'
  },
  {
    _id: ObjectId("65212b5cab4dfc111e000350"),
    Rno: '6',
    Name: 'Nikhil',
    Class: 'TE COMPUTER'
  }
]

```

```
AK> db.Student.find( {"Rno": { $gt:"5"}}).pretty();
```

```
[
  {
    _id: ObjectId("65212b5cab4dfc111e000350"),
    Rno: '6',
    Name: 'Nikhil',
    Class: 'TE COMPUTER'
  }
]
```

```
AK> db.Student.find( {"Rno": { $gte:"5"}}).pretty();
```

```
[
  {
    _id: ObjectId("651a4e7fe64b91d30b77bd86"),
    Rno: '5',
    Name: 'Nikhil',
    Class: 'TE Computer'
  },
  {
    _id: ObjectId("65212b5cab4dfc111e000350"),
    Rno: '6',
    Name: 'Nikhil',
    Class: 'TE COMPUTER'
  }
]
```

```
AK> db.Student.find( {"Rno": { $lt:"5"}}).pretty();
```

```
[
  {
    _id: ObjectId("651a4d4fe64b91d30b77bd80"),
    Rno: '1',
    Name: 'Atharva',
    Class: 'TE Computer'
  },
  {
    _id: ObjectId("651a4d4fe64b91d30b77bd81"),
    Rno: '2',
    Name: 'Shubham',
    Class: 'TE Computer'
  },
  {
    _id: ObjectId("651a4d4fe64b91d30b77bd82"),
    Rno: '3',
    Name: 'Yash',
    Class: 'TE Computer'
  },
  {
    _id: ObjectId("651a4d4fe64b91d30b77bd83"),
    Rno: '4',
    Name: 'Nikita',
    Class: 'TE Computer'
  }
]
```

```
AK> db.Student.find( {"Rno": { $lt:"5",$gt:"2"}}).pretty();
```

```
[
  {
    _id: ObjectId("651a4d4fe64b91d30b77bd82"),
    Rno: '3',
    Name: 'Yash',
    Class: 'TE Computer'
  },
  {
    _id: ObjectId("651a4d4fe64b91d30b77bd83"),
    Rno: '4',
    Name: 'Nikita',
    Class: 'TE Computer'
  }
]
```

```
AK> db.Student.find( {"Rno": { $lte:"5",$gte:"2"}}).pretty();
```

```
[
  {
    _id: ObjectId("651a4d4fe64b91d30b77bd81"),
    Rno: '2',
    Name: 'Shubham',
    Class: 'TE Computer'
  },
  {
    _id: ObjectId("651a4d4fe64b91d30b77bd82"),
    Rno: '3',
    Name: 'Yash',
    Class: 'TE Computer'
  },
  {
    _id: ObjectId("651a4d4fe64b91d30b77bd83"),
    Rno: '4',
    Name: 'Nikita',
    Class: 'TE Computer'
  },
  {
    _id: ObjectId("651a4e7fe64b91d30b77bd86"),
    Rno: '5',
    Name: 'Nikhil',
    Class: 'TE Computer'
  }
]
```

DBMS Assignment No.10

Please enter a MongoDB connection string (Default: `mongodb://localhost/`): `mongodb:mongodb:`
Current Mongosh Log ID: 651a5bb6d2cc8e767d9fb00a
Connecting :
`mongodb://127.0.0.1:27017/mongodb?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.10.6`
Using MongoDB: 7.0.1 Using Mongosh: 1.10.6
mongosh 2.0.1 is available for download: <https://www.mongodb.com/try/download/shell>
For mongosh info see: <https://docs.mongodb.com/mongodb-shell/>

The server generated these startup warnings when booting
2023-09-30T22:09:56.935+05:30: Access control is **not** enabled for the database.
Read and write access to data and configuration is unrestricted

```
mongodb> show dbs
AK           72.00 KiB
Atharva      8.00 KiB
admin        40.00 KiB
config       108.00 KiB
local        40.00 KiB
```

```
mongodb> use AK
switched to db AK
```

```
AK> db.createCollection('website');
{ ok: 1 }
```

```
AK> db.website.insertMany
  ([{'Roll': '1', 'Name': 'Atharva', 'Amount': 1000, 'URL': 'www.yahoo.com'},
    {'Roll': '2', 'Name': 'Shubham', 'Amount': 2000, 'URL': 'www.yahoo.com'},
    {'Roll': '3', 'Name': 'Nikhil', 'Amount': 3000, 'URL': 'www.google.com'},
    {'Roll': '4', 'Name': 'Nikita', 'Amount': 4000, 'URL': 'www.gmail.com'},
    {'Roll': '5', 'Name': 'Nikita', 'Amount': 1000, 'URL': 'www.pvg.com'}]);
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("651a5c17d2cc8e767d9fb00b"),
    '1': ObjectId("651a5c17d2cc8e767d9fb00c"),
    '2': ObjectId("651a5c17d2cc8e767d9fb00d"),
    '3': ObjectId("651a5c17d2cc8e767d9fb00e"),
    '4': ObjectId("651a5c17d2cc8e767d9fb00f")
  }
}
```

```
AK> db.website.aggregate({'$group':{'_id':'$Name','Total':{'$sum':'$Amount'}}});
[
  { _id: 'Nikhil', Total: 3000 },
  { _id: 'Nikita', Total: 5000 },
  { _id: 'Atharva', Total: 1000 },
  { _id: 'Shubham', Total: 2000 }
]
```

```
AK> db.website.aggregate({$group:{_id:"$Name","Total":{$avg:"$Amount"}}});
[
  { _id: 'Nikhil', Total: 3000 },
  { _id: 'Nikita', Total: 2500 },
  { _id: 'Atharva', Total: 1000 },
  { _id: 'Shubham', Total: 2000 }
]
```

```
AK> db.website.aggregate({$group:{_id:"$Name","Total":{$min:"$Amount"}}});
[
  { _id: 'Atharva', Total: 1000 },
  { _id: 'Nikhil', Total: 3000 },
  { _id: 'Nikita', Total: 1000 },
  { _id: 'Shubham', Total: 2000 }
]
```

```
AK> db.website.aggregate({$group:{_id:"$Name","Total":{$max:"$Amount"}}});
[
  { _id: 'Atharva', Total: 1000 },
  { _id: 'Nikhil', Total: 3000 },
  { _id: 'Nikita', Total: 4000 },
  { _id: 'Shubham', Total: 2000 }
]
```

```
AK> db.website.aggregate({$group:{_id:"$Name","Total":{$first:"$Amount"}}});
[
  { _id: 'Atharva', Total: 1000 },
  { _id: 'Nikhil', Total: 3000 },
  { _id: 'Nikita', Total: 4000 },
  { _id: 'Shubham', Total: 2000 }
]
```

```
AK> db.website.aggregate({$group:{_id:"$Name","Total":{$last:"$Amount"}}});
[
  { _id: 'Nikhil', Total: 3000 },
  { _id: 'Shubham', Total: 2000 },
  { _id: 'Atharva', Total: 1000 },
  { _id: 'Nikita', Total: 1000 }
]
```

```
AK> db.website.aggregate({$group:{_id:"$Name","Total":{$push:"$Amount"}}});
[
  { _id: 'Shubham', Total: [ 2000 ] },
  { _id: 'Atharva', Total: [ 1000 ] },
  { _id: 'Nikita', Total: [ 4000, 1000 ] },
  { _id: 'Nikhil', Total: [ 3000 ] }
]
```

```
AK> db.website.aggregate({$group:{_id:"$Name","Total":{$sum:1}}});
[
  { _id: 'Atharva', Total: 1 },
  { _id: 'Nikhil', Total: 1 },
  { _id: 'Nikita', Total: 2 },
  { _id: 'Shubham', Total: 1 }
]
```

```

AK> db.website.aggregate({$group:{_id:"$Name","Total":{$addToSet:"$Amount"}}});
[
  { _id: 'Atharva', Total: [ 1000 ] },
  { _id: 'Nikhil', Total: [ 3000 ] },
  { _id: 'Nikita', Total: [ 4000, 1000 ] },
  { _id: 'Shubham', Total: [ 2000 ] }
]

AK> db.createCollection('website1');
{ ok: 1 }
AK> db.website1.insertOne({'r':1,'name':'xyz'});
{
  acknowledged: true,
  insertedId: ObjectId("651a61a9d2cc8e767d9fb010")
}
AK> db.website1.find().pretty();
[ { _id: ObjectId("651a61a9d2cc8e767d9fb010"), r: 1, name: 'xyz' } ]

AK> db.website1.createIndex({'name':1})
name_1
AK> db.website1.createIndex({'name':-1})
name_-1

AK> db.website1.getIndexes()
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { name: 1 }, name: 'name_1' },
  { v: 2, key: { name: -1 }, name: 'name_-1' }
]

AK> db.website1.createIndex({'name':2});
name_2
AK> db.website1.getIndexes();
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { name: 1 }, name: 'name_1' },
  { v: 2, key: { name: -1 }, name: 'name_-1' },
  { v: 2, key: { name: 2 }, name: 'name_2' }
]

AK> db.website.dropIndex({'name':-1});
MongoshInternalError: can't find index with key: { name: -1 }

```


DBMS Assignment no.11

Please enter a MongoDB connection string (Default: `mongodb://localhost/`): `mmongodb:`
`mmongodb:`
Current Mongosh Log ID: 651a755fec0b877145512c93
Connecting to:
`mongodb://127.0.0.1:27017/mmongodb?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.10.6`
Using MongoDB: 7.0.1
Using Mongosh: 1.10.6
mongosh 2.0.1 is available for download: <https://www.mongodb.com/try/download/shell>
For mongosh info see: <https://docs.mongodb.com/mongosh-shell/>

The server generated these startup warnings when booting
`2023-09-30T22:09:56.935+05:30: Access control is not enabled for the database.`
Read and write access to data and configuration is unrestricted

```
mmongodb> show dbs
AK          212.00 KiB
Atharva     8.00 KiB
admin       40.00 KiB
config      84.00 KiB
local       40.00 KiB
```

```
mmongodb> use AK
switched to db AK
```

```
AK> db.createCollection('Journal');
{ ok: 1 }
```

```
AK>
db.Journal.insertMany([{'book_id':1,'book_name':'JavacdOOP','amt':500,'status':'Available'},
  {'book_id':1,'book_name':'JavaOOP','amt':400,'status':'NotAvailable'},
  {'book_id':1,'book_name':'Java','amt':300,'status':'NotAvailable'},
  {'book_id':2,'book_name':'Java','amt':300,'status':'Available'},
  {'book_id':2,'book_name':'OOP','amt':200,'status':'Available'},
  {'book_id':2,'book_name':'C++','amt':200,'status':'Available'},
  {'book_id':3,'book_name':'C++','amt':150,'status':'Available'},
  {'book_id':3,'book_name':'C++','amt':200,'status':'NotAvailable'},
  {'book_id':4,'book_name':'OOP C++','amt':300,'status':'NotAvailable'},
  {'book_id':5,'book_name':'OOP C++','amt':400,'status':'Available'},
  {'book_id':5,'book_name':'C++','amt':400,'status':'Available'},
  {'book_id':5,'book_name':'C++ Java','amt':400,'status':'NotAvailable'}]);
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("651a7780ec0b877145512c94"),
    '1': ObjectId("651a7780ec0b877145512c95"),
    '2': ObjectId("651a7780ec0b877145512c96"),
    '3': ObjectId("651a7780ec0b877145512c97"),
    '4': ObjectId("651a7780ec0b877145512c98"),
    '5': ObjectId("651a7780ec0b877145512c99"),
```

```

    '6': ObjectId("651a7780ec0b877145512c9a"),
    '7': ObjectId("651a7780ec0b877145512c9b"),
    '8': ObjectId("651a7780ec0b877145512c9c"),
    '9': ObjectId("651a7780ec0b877145512c9d"),
    '10': ObjectId("651a7780ec0b877145512c9e"),
    '11': ObjectId("651a7780ec0b877145512c9f")
  }
}
AK> var mapfunction=function(){ emit(this.book_id,this.amt)};

AK> var reducefunction=function(key,value){return Array.sum(value)};

AK> db.Journal.mapReduce(mapfunction,reducefunction,{ 'out': 'new' });
{ result: 'new', ok: 1 }

AK> db.new.find().pretty();
[
  { _id: 3, value: 350 },
  { _id: 4, value: 300 },
  { _id: 5, value: 1200 },
  { _id: 2, value: 700 },
  { _id: 1, value: 1200 }
]

```

DBMS Assignment No. 12

```
import java.net.UnknownHostException;
import java.util.Scanner;
import com.mongodb.*;

public class DatabaseConnectivity
{
    private static void choice_input()
    {
        System.out.println("\n1.insert data into database\n2.update database
documents\n3.delete database documents\n4.show database collections\n5.Exit");
    }

    public static void main(String[] args)
    {
        String key, value;
        Scanner scanner = new Scanner(System.in);
        int choice;
        try
        {
            Mongo mongo = new Mongo("localhost", 27017);
            DB db = mongo.getDB("myDb");
            DBCollection collection = db.getCollection("dummyColl");
            do
            {
                choice_input();
                System.out.println("Enter your choice: ");
                choice = scanner.nextInt();switch (choice)
                {
                    case 1:
                        BasicDBObject document = new BasicDBObject();
                        String ch;
                        do
                        {
                            System.out.println("Enter key: ");
                            key = scanner.next();
                            System.out.println("Enter value: ");
                            value = scanner.next();
                            document.put(key, value);
                            System.out.println("Do you want to enter more(y/n)? ");
                            ch = scanner.next();
                        }
                        while (!ch.equals("n"));
                        collection.insert(document);
                        break;

                    case 2:
                        BasicDBObject searchObj = new BasicDBObject();
                        System.out.println("Enter searched key: ");
                        key = scanner.next();
                        System.out.println("Enter searched value: ");
                        value = scanner.next();
                        searchObj.put(key, value);
```

```

        BasicDBObject newObj = new BasicDBObject();
        System.out.println("Enter new key: ");
        key = scanner.next();
        System.out.println("Enter new value: ");
        value = scanner.next();
        newObj.put(key, value);
        collection.update(searchObj, newObj);
    break;

    case 3:
        System.out.println("Enter removable key: ");
        key = scanner.next();
        System.out.println("Enter removable value: ");
        value = scanner.next();
        BasicDBObject removableObj = new BasicDBObject();
        removableObj.put(key, value);
        collection.remove(removableObj);
    break;

    case 4:
        DBCursor cursorDoc = collection.find();
        while (cursorDoc.hasNext())
        {
            System.out.println(cursorDoc.next());
        }
    break;

    case 5:
        System.exit(0);
    break;
}
}
while(choice != 6);
}
catch (UnknownHostException | MongoException e)
{
    e.printStackTrace();
}
}
}

```

Output:

```

1.insert data into database
2.update database documents
3.delete database documents
4.show database collections
5.Exit

```

Enter your choice:

1

Enter key:2

Enter value:

harish

Do you want to enter more(y/n)?

N

1.insert data into database
2.update database documents
3.delete database documents
4.show database collections
5.Exit

Enter your choice:

2

Enter searched key:

2

Enter searched value:

harish

Enter new key:

1

Enter new value:

Sam

1.insert data into database
2.update database documents
3.delete database documents
4.show database collections
5.Exit

Enter your choice:

4

```
{"_id" : { "$oid" : "5bb453bce4b0283ac9d3205d" } , "1" : "sam"}
```

1.insert data into database
2.update database documents
3.delete database documents
4.show database collections
5.Exit

Enter your choice:

3

Enter removable key:

3

Enter removable value:

hari

1.insert data into database
2.update database documents
3.delete database documents
4.show database collections
5.Exit

Enter your choice:

4

```
{"_id" : { "$oid" : "5bb453bce4b0283ac9d3205d" } , "1" : "sam"}
```

1.insert data into database
2.update database documents
3.delete database documents
4.show database collections
5.Exit

Enter your choice:

5